Quintana, D., Cervantes, A., Saez, Y. et al. Clustering technique for large-scale home care crew scheduling problems. Appl Intell 47, 443–455 (2017).

DOI: https://doi.org/10.1007/s10489-017-0908-1

Applied Intelligence manuscript No.
(will be inserted by the editor)

Clustering technique for large-scale home care crew scheduling problems

David Quintana · Alejandro Cervantes · Yago Sáez · Pedro Isasi

Received: date / Accepted: date

Abstract The Home Health Care Scheduling Problem requires allocating professional caregivers to patient's place of residence to meet service demands. These services are regular in nature and must be provided at specific times during the week. In this paper we present a heuristic with two tie breaking mechanisms suitable for large-scale versions of the problem. The greedy algorithm merges service lots minimizing accumulated non-service time and, as a result, restructures the solution increasing its efficiency. The approach is tested on a real-world large instance of the problem for a company whose current resource allocation is inefficient. The solutions are benchmarked against the current service assignment and those obtained by a Ward clustering algorithm, and the results show an improvement in efficiency.

Keywords Scheduling · Home Health Care · Clustering · Heuristics

The authors acknowledge financial support granted by the Spanish Ministry of Science under contract TIN2011-28336 (MOVES).

D. Quintana
Dept. of Computer Science
Universidad Carlos III de Madrid
Av. de la Universidad 30
Leganes, Madrid 28911, Spain
Tel.: +34-916249109
Fax: +34-916249430
E-mail: dquintan@inf.uc3m.es

A. Cervantes
E-mail: acervant@inf.uc3m.es

Y. Saez
E-mail: ysaez@inf.uc3m.es

P. Isasi
E-mail: isasi@ia.uc3m.es

1 Introduction

Home Health Care (HHC) or domiciliary care is the business of providing professional assistance services in the patient's place of residence according to a formal assessment of their needs. The patient's needs can vary from health-care like nursing, medical treatments, physical therapy, speech therapy to home assistance services like cleaning, dressing, cooking, etc. The aim of HHC is to provide the care and support needed to assist people, particularly the elderly, population with learning or physical disabilities and people who need assistance due to illness and still want to live as independently as possible in their own homes. In this context, customers usually prefer to be attended by the same caregiver so that a familiar and confidence relationship starts being developed from the beginning of the service.

Nowadays Home Health Care services are becoming a growing business and an important issue in Europe's aging societies. According to EUROSTAT "EU-27's old-age dependency ratio, defined as the population aged 65 years and older divided by the population aged 15 to 64, is projected to more than double from 26.2% in 2011 to 52.6% by 2060" [1]. As the number of customer increases, the complexity of the HHC operations planning increases too, and HHC providers must find new ways to remain profitable, optimizing operational costs while keeping good quality of service. As it is a growing business, it has been receiving recent attention from researchers [2–11]

The cornerstone of HHC providers is to optimize operational costs while keeping high quality service. Two main components contribute to operational cost: overtime and length of tours. This means that the operational planning made HHC providers tries to minimize these cost factors while providing a good level service to the customers, e.g seeing them at their desired time of the day by a caregiver who they know well and trust.

Operational planning at HHC providers is nowadays mostly done manually, often by experienced senior caregivers. The planning problem is quite complex. In order to tackle it manually, HHC providers usually work with a mid-term plan that is the template used for day-to-day operational planning. Daily modifications are made to adjust to occasional unavailability of caregivers or temporary changes in the tasks to be performed. The mid term plan is updated less frequently, mainly to consider new customers or customers that have decided leave the service.

In this work, we introduce a clustering heuristic with two tie breaking mechanisms to address a real-world large-scale example of the mentioned mid term planning. The instance of the planning problem, known as the Home Care Crew Scheduling Problem (HCCSP), was developed in collaboration with a company that, at the time this research, was suffering the burden of an inefficient allocation plan for their services in Madrid, and decided to explore alternatives.

Since it was proposed by [12], the Home Care Crew Scheduling Problem has been addressed and formulated in several ways. In its more complex form, customer and caregivers should be matched according to requirements and skills. Customers may require multiples services with several skills requiring that caregiver's visits must be synchronized, and customers must be attended within certain time windows. Caregivers have several working turns such as full-time, part-time or split-shift that must be respected including lunch break. With this perspective, HCCSP is NP-hard problem [13], since its a combination of two NP-Hard problems, the vehicle routing problem with time windows and the nurse rostering problem.

HCCSP has been addressed in the literature a number of times. In [14] authors propose hybridizing constraint programming with meta-heuristics, namely, simulated annealing and tabu search. The model is solved in two phases; in the first stage, the set of services are partitioned in sets that have to be performed by a single nurse. Afterwards, an order is generated for every service. Using this approach, the authors were able to deal with synthetic test instances of 20-30 nurses with 200-600 services to be scheduled.

HCCSP has also been modeled as a partitioning problem consisting of visits and staff members. This is the approach proposed by [15], where the final aim is to match visits to staff members in such way that the constraints are satisfied. Using a repeated matching algorithm [16] authors solved instances with up to 20 members and 123 visits which are solved within 140 seconds. In [17] the daily HCCSP is addressed as a vehicle routing problem with time windows and solved it with the particle swarm optimization meta-heuristics. The objective consists of minimizing the total distance traveled, while meeting constraints related to time windows, operators and patients. Rend et at. [18] test several hybrid solution methods for a real-world instance of HCCS where a number of nurses travel in tours of using different modes of transport. The strategies start with basic solutions obtained with combination of Constraint Programming and an iterative clustering algorithm based on Quadtrees which are subsequently refined with heuristics.The instances considered by this authors include about 700 jobs and 500 available nurses.

More recently, researchers have tackled cardinality-constrained robust strategies [19], while Maya Duque et al.[20] studied a home care planning problem using a two-phase algorithm. In this case, the first stage of the solution strategy optimizes the service level, while the second aims to minimize the total distance traveled by caregivers.

Reuven Levary [21] minimizes the number of caregivers on nurse home care scheduling problem using heuristics, while the authors of [22] describe a constraint programming model to optimize an instance in Ferrara balancing objectives regarding workload or travel times among others.

The clustering approach that we present aims at improving the efficiency of the allocation of caregivers to services. The method is tested addressing a HCCSP instance for Madrid. Unlike most of the real-world application examples on the topic found in the literature, this one is characterized by its scale. As we will discuss in the next pages, the number of tasks to be allocated lies in the thousands.

The rest paper is structured as follows. The next section describes in detail the specifics the instance. That will be followed by the introduction of the algorithmic approach used to tackle the problem. Then, we present the experimental results and, finally, the last section is devoted to conclusions and future work.

2 Instance Description

In this work, we face a, large scale, real world, home care scheduling problem, in the city of Madrid, where 9,635 patients must be visited periodically, at least once a week, at their homes. Given the sensitive nature of the matter, the search of solutions should consider the following guidelines:

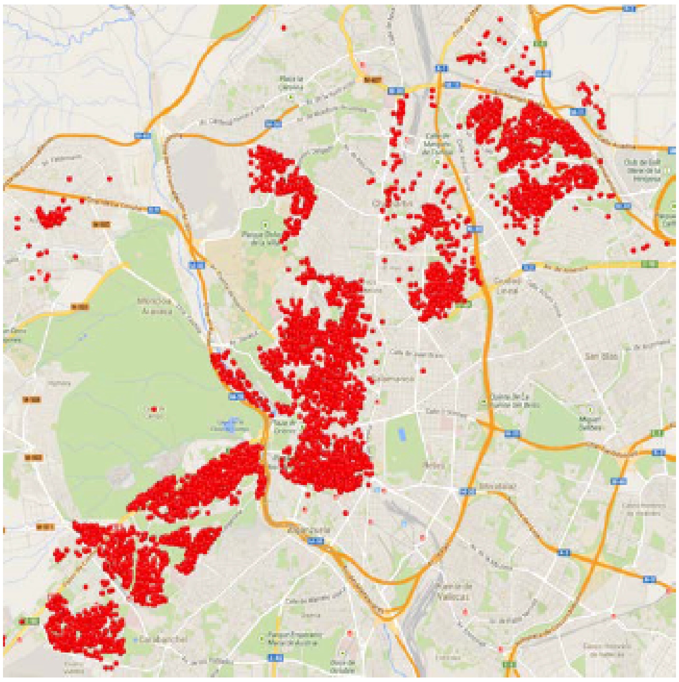

Fig. 1 Tasks positioned over city map. Each dot represents a task.

- The quality of the service must be maximized.
- Patients should be served on time and regularly by the same team to allow trust and confidence between caregivers and patients.
- Working conditions of caregivers must be taken into account and respected.
- Service costs, supported by the municipality, should be minimized to maximize the number of patients potentially served with a fixed budged.

These features, together with the scale of the problem, make this home care scheduling task somehow particular and do not allow the direct use of regular scheduling techniques. It requires ad hoc solutions or at least the adaptation of known techniques to incorporate the particularities of the problem.

One of the main characteristics of the home care scheduling problem in big cities is scale. We found that, in this particular situation, 29,034 tasks in the patient's home must be assigned. These include, among others, managing laundry, providing medication, cleaning, etc. The nature of all them is repetitive, and they begin at a specific time and have a specific duration.

The need to limit the number of different caregivers that visit a patient and the requirement to consider specific work shifts resulted in the creation of 13.344 task sets, or services. All the tasks in the same service should be assigned to the same caregiver and, therefore, should be considered an atomic structure from an allocation point of view.

Picture 1 shows the total number of services being scheduled, whereas Picture 2 shows the set of tasks clustered using a simple grid clustering algorithm that uses geographic distance.

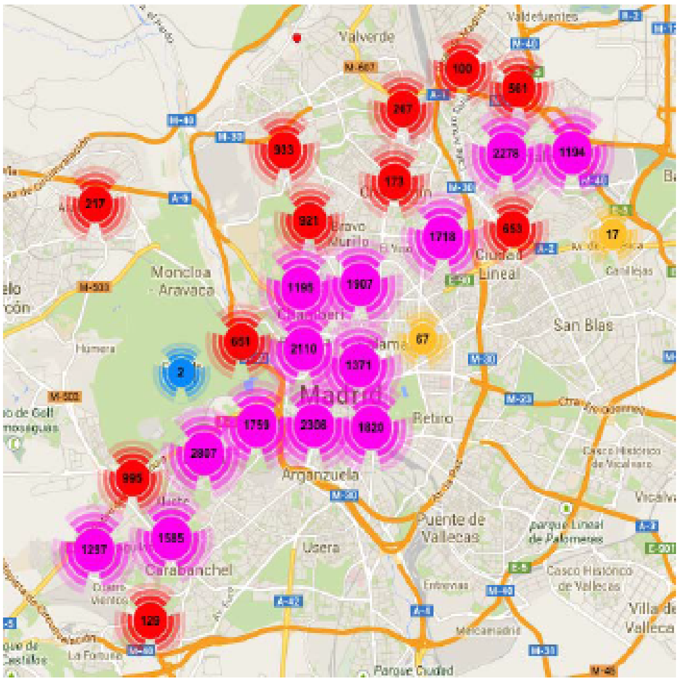

Fig. 2 Projection of clustered tasks. Note that each cluster has been formed using a grid and therefore constraints have not been taken into account.

Table 1 Problem example with three services

Service	Task	Day (0-6)	Start α_i	Duration d_i	Location $(lat_i, long_i)$
134	1	Monday	10:30	80	(-3.709, 40.425)
	2	Friday	10:45	65	(-3.709, 40.425)
237	1	Monday	13:30	40	(-3.772, 40.374)
457	1	Monday	13:00	60	(-3.748, 40.396)
	2	Tuesday	11:00	60	(-3.748, 40.396)
	3	Friday	11:00	60	(-3.748, 40.396)

In Table 1 we describe an example problem with only three services. Each task i is defined as a vector $\{\alpha_i, d_i, x_i, y_i\}$, with a start time (α_i), a duration (d_i), and the geographical location of the customer $(lat_i, long_i)$.

A **solution** for this problem is a set of assignments of services to caregivers, that define an itinerary or path to be followed by each caregiver every day.

Solutions must be **feasible**, that is, it must meet several constraints: a) tasks may not overlap in time; b) customer locations must be close enough for the caregiver to service one task and move to the location of the next task on time; c) caregivers work in one of three shifts: weekday mornings, weekday afternoons or weekends. That means, for instance, that a caregiver cannot be assigned to services

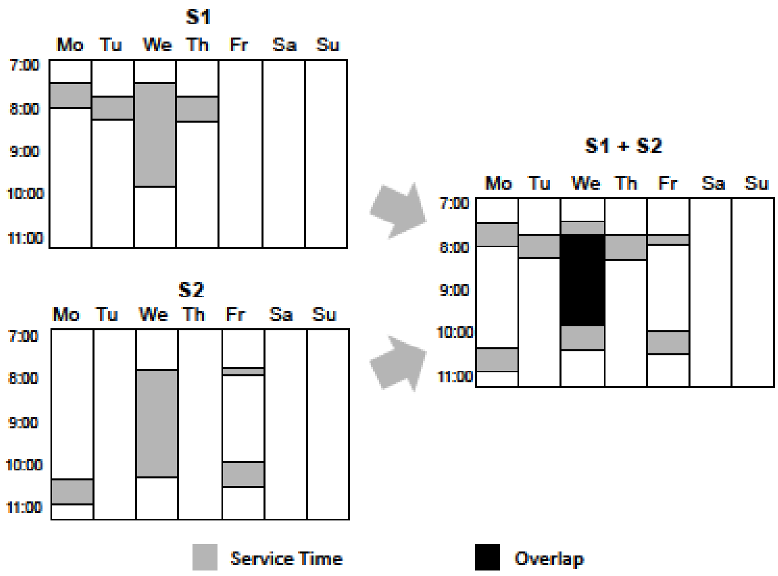

Fig. 3 Illustration of task overlap for two services.

both in the morning and the afternoon during weekdays; and d) caregivers cannot exceed a maximum working time per week (40 hrs if full-time, 20 hrs if part-time).

Travel times are based on walking distances among service locations. For this purpose, the computation of distances is direct. We just consider differences in latitude and longitude. No mapping software was used to fine-tune the estimates.

A fact of critical importance to understand the complexity of the problem is that feasibility has to be calculated over the whole week. An assignment of a service to a caregiver may be correct for a given day, but it would be unfeasible if it assigns conflicting tasks for a different day of the week, as illustrated in Fig. 3.

Looking at the detailed example presented in Table 2, it can be proved that at least two different caregivers are required for any feasible solution: services 134 and 457 are incompatible due to the overlap between tasks 134.2 and 457.3, and service 237 is incompatible with service 457 due to the overlap between tasks 237.1 and 457.1.

In Table. 2 we show a feasible solution for the services in Table 1, where caregivers are identified as #321 and #45. Note that in order to perform tasks 134.1 and 237.1, caregiver #321 has to spend 90 minutes waiting for task 237.1 to start. This idle time ($w_{ij}^{321,0}$) is very undesirable, as the caregiver per-week time includes these idle times. However this is the only feasible two-caregivers solution.

As we mentioned earlier, a solution is an allocation of caregivers to services in such a way that no service is left unattended. This is equivalent to a search for a partition of the space of services, where each cluster corresponds to the services assigned to the same caregiver. More formally, the problem can be framed as: given a set of services $S = \{s_1..., s_j, ..., s_n\}$, the aim is to find a partition of S,

Table 2 Solution example for two caregivers: #321 and #45. Tasks are identified as (`Service.Task`). For Monday we show the working period in dark grey, travel time in light grey. Times (in minutes) used for solution evaluation are shown only for Monday

Caregiver #321					
Time	Monday		Tuesday	...	Friday
10:30					
10:45	134.1	$t_i^{321,0} = 80$			134.2
11:49					
11:50	Traveling	$d_{ij}^{321,0} = 10$			
12:00	Waiting	$w_{ij}^{321,0} = 90$			
13:30	237.1	$t_j^{321,0} = 40$			
14:10					
Caregiver #45					
Time	Monday		Tuesday	...	Friday
11:00			457.2		457.3
11:59					
13:00	457.1	$t_i^{45,0} = 60$			
13:59					

Table 3 Solution evaluation for the assignment in Table 2

Caregiver #321				
Time	Monday	Tuesday	Friday	**Totals**
Start	10:30		10:45	
End	14:10		11:50	
Tasks	$\sum t_i = 120$		$\sum t_i = 65$	$TT_{321} = 185$
Travel	$\sum d_{ij} = 10$		$\sum d_{ij} = 0$	$DT_{321} = 10$
Waiting	$\sum w_{ij} = 90$		$\sum w_{ij} = 0$	$WT_{321} = 90$
Caregiver #45				
Time	Monday	Tuesday	Friday	**Totals**
Start	13:00	11:00	11:00	
End	14:00	12:00	12:00	
Work	$\sum t_i = 60$	$\sum t_i = 60$	$\sum t_i = 60$	$TT_{45} = 180$
Travel	$\sum d_{ij} = 0$	$\sum d_{ij} = 0$	$\sum d_{ij} = 0$	$DT_{45} = 0$
Waiting	$\sum w_{ij} = 0$	$\sum w_{ij} = 0$	$\sum w_{ij} = 0$	$WT_{45} = 0$
Solution evaluation measures				
$TT = \sum_{K^*} TT_k = 365$		$DT = \sum_{K^*} DT_k = 10$		
$WT = \sum_{K^*} WT_k = 90$		$T = TT + DT + WT = 465$		

$C = \{C_1..., C_K\}(K \leq N)$, such that

$$C_i \neq \emptyset, i = 1, .., K \tag{1}$$

$$\bigcup_{i}^{K} C_i = S \tag{2}$$

$$C_i \cap C_j = \emptyset, i, j = 1, .., K, j \neq i \tag{3}$$

In Table 3 we describe how we perform solution evaluation. For each cluster we calculate the following values: Task Time (TT_k), Travel Time (DT_k) and Waiting Time (WT_k). Total time for a cluster (T_k) equals $TT_k + DT_k + WT_k$, and corresponds to the full periods of time that span from each day's start of work (start time for the first task) until the day's end of work (end time for the last task of the day).The solution can be evaluated using the same measures, by totalling these measures for all clusters, giving a solution total, task, travel and waiting times (T,TT,DT and WT).

The problem has a multi-objective nature with three conflicting objectives that have to be minimized:

1. Total number of clusters $|K|$
2. Total travel time (DT):

$$\mathrm{DT} = \sum_{\forall d \in D} \sum_{\forall k \in K^*} \sum_{\forall i, j=i+1} d_{ij}^{k,d} \tag{4}$$

3. Total waiting time (WT):

$$\mathrm{WT} = \sum_{\forall d \in D} \sum_{\forall k \in K^*} \sum_{\forall i, j=i+1} w_{ij}^{k,d} \tag{5}$$

In Eqs. 4 and 5, D is the set of days (from Monday to Sunday, coded 0 to 6), and the indices i and i, j designate tasks: for instance, $d_{ij}^{k,d}$ is the time to travel between the locations in two *consecutive* tasks i and $j = i + 1$ that are both in cluster k and are both performed in day d. Similarly, $w_{ij}^{k,d}$ is the waiting time between those same tasks.

It is obvious that times (both DT and WT) can be reduced by increasing the number of caregivers (that is, the number of clusters $|K|$). For instance, in the given example (Table 1 and Table 2), the company might be able to increase its staff with a third caregiver. If that extra person is assigned service 231 we would achieve optimal service times ($DT = 0$ and $WT = 0$). However the final cost of the new solution may be higher, depending on the marginal cost of hiring a new caregiver. The reverse is also true: as marginal costs increase, better solutions are probably solutions with lower $|K|$ even when DT and WT increase.

As previously mentioned, our purpose is to define a constructive algorithm that builds a solution from the defined services. However, we must stress that this problem has additional complexity compared to previous work in literature. This is the reason why the task cannot be performed efficiently by standard clustering techniques:

- Clustering algorithms distances have to take into account not only geographical distance between task locations, but also the time dimension (spatio-temporal clustering).
- We have to take into account four feasibility constraints: a) task compatibility, b) travel time, c) time shift and d) maximum per-week work time.

– As services contain tasks that are performed in different days, constraints have to be evaluated over the whole week.
– Size of the problem is unusually large.

As a result, we have studied several approaches that involve a special treatment of distances and solution feasibility. Also we introduce some relaxations in the aforementioned constraints that seem consistent with the nature of the problem.

3 Algorithmic Approach

In this section we present the merger heuristic with two tie breaking mechanisms used to tackle the problem.

The approach that we suggest is an agglomerative greedy clustering algorithm that creates compact services sets. The process can be succinctly described as follows. The algorithm starts considering each service as a cluster, C_s. Then, it explores all the potential two-cluster mergers that comply with the constraints. In case there is more than one potential combination, the algorithm will pick the most desirable one. The process iterates, one merger at a time, until it runs out of feasible combinations. The resulting collection of clusters, or service sets, are the suggested solution.

Figure 4 represents a cluster of three services that could potentially be obtained in the process. In this case, service 1 consists of two tasks that must be performed on Mondays and Wednesdays in the morning. Service 2 requires three morning tasks on Tuesdays, Thursdays and Saturdays. Finally, service 3 involves a single short task on Thursdays afternoon. Given that there are two consecutive tasks on Thursdays, the caregiver will face a 5-hour idle time period.

A potential merger is desirable if it is both feasible, and the combination of the pair of clusters results in a schedule where the accumulated idle time for week is low. Conversely, if the resulting schedule is not compact, the weekly amount of idle time is high, the merger will not be considered attractive.

Figure 5 exemplifies this concept. In this case, the situation involves three initial clusters C_1, C_2 and C_3 that consist of 5, 5 and 4 tasks respectively. These tasks are distributed during the week and might be part of a single service, or the result of the aggregation of several services though cluster combination in previous iterations of the algorithm. There are three potential cluster combinations, C_1+C_2, C_2+C_3 and C_1+C_3, that are illustrated in the second row. Out of these, the first one is unfeasible due to the evident task overlap on Wednesday morning. The remaining two do not suffer from this limitation and therefore, provided that they meet the rest of the constraints, are the only real alternatives. Out of these, the heuristic would consider C_1+C_3 the most promising. This is due to the fact that it leads to the smallest accumulated idle time. The solution is more compact, hence, the *Preferred* label.

At this point, we have the elements to provide a more detailed description of the algorithm. Initially, all the services are identified as clusters of size one. Each of these would consist of one or more tasks that should necessarily be undertaken by the same caregiver. Then, we initialize the cost table, M. This symmetric double entry table stores for every potential combination of two existing service-sets, identified by row and column, the weekly idle time resulting from their combination. When a merger is not possible, due to either task overlap, insufficient time

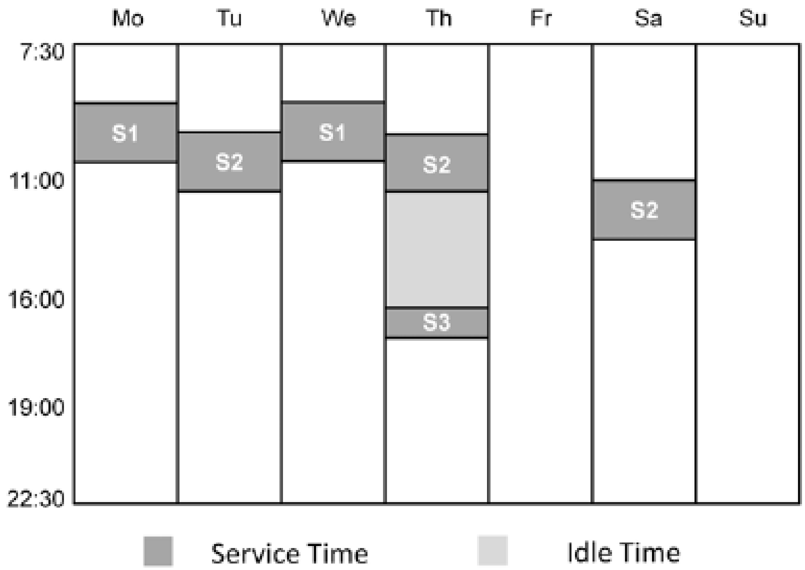

Fig. 4 Graphic representation of a cluster involving six tasks grouped in three services

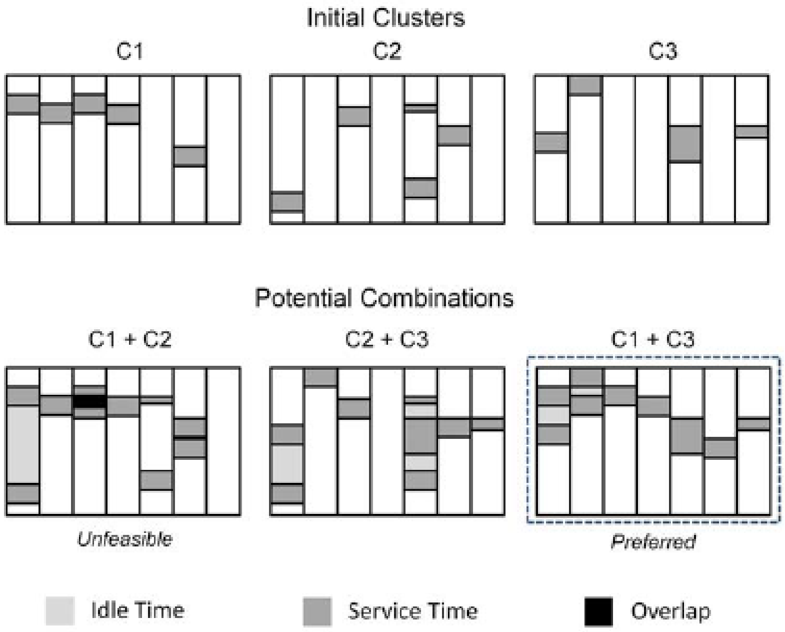

Fig. 5 Merger attractiveness. Example for three clusters C_1, C_2 and C_3 showing the three potential combinations.

	C1	C2	C3	C4	C5	C6	C7	C8
C1		10		15	55	50	40	20
C2	10		20	50	70	80	90	30
C3		20		25	85	30	65	
C4	15	50	25		50		70	10
C5	55	70	85	50		5	80	60
C6	50	80		30	5		40	25
C7	40	90	65	70	80	40		30
C8	20	30		10	60	25	30	

Fig. 6 Illustration of matrix M. Initial state.

between two consecutive tasks to transfer or inappropriate work schedules, the cell at the row/column intersection for pair of clusters gets labeled as *unfeasible*.

Figure 6 shows an example of matrix M based on eight initial clusters. The number at the intersection of rows and columns represents the cost of the combination and, in case the combination violates the mentioned constraints, the cell is colored in dark grey. In this case, the aggregation of clusters $C_2 + C_3$ has a cost of 20, which is a better alternative than C_2 and C_7, which results in a weekly accumulated idle time of 90. The combination of $C_1 + C_3$ or $C_3 + C_8$ would be unfeasible, and so would be all the combinations of any cluster with itself (the main diagonal).

We should mention that algorithm rounds traveling times to the nearest multiple of five minutes, therefore allowing a maximum of a 2.5 minute overlap. This results on a slight relaxation of the mentioned constraint on transfer times between consecutive tasks. In terms of values in the table, the smaller the value, the more attractive the combination. Once the previous initialization is carried out, we get to the main loop.

Iterations begin with the selection of the smallest feasible element in M. This represents the merger of the two existing clusters that leads to lowest accumulated weekly idle time. In the example, this would be the combination of clusters 5 and 6. Once these elements have been identified, the set identified by the column is merged into the set identified by the row. Finally, the cost table, M, gets updated.

This last step requires two tasks. On one hand, labeling every potential combination of the service set in the column as unfeasible (for all purposes, the cluster does not exist anymore). On the other hand, computing the cost of combining the new cluster with the rest of the remaining services sets, and updating with this information the figures from the old set that was previously identified by the row number. This loop is repeated while there are feasible combinations, that is, while there is at least an element in the table other than an *unfeasible* label.

	C1	C2	C3	C4	C5+C6	C6	C7	C8
C1		10		15	35		40	20
C2	10		20	50	60		90	30
C3		20		25	85		65	
C4	15	50	25		40		70	10
C5+C6	35	60	85	40			95	60
C6								
C7	40	90	65	70	95			30
C8	20	30		10	60		30	

Fig. 7 Illustration of matrix M. Intermediate state.

The result of the execution of the algorithm is a set of non-empty clusters of services $C = \{C_1..., C_K\}$ that balances the number of required care givers, and the accumulated non-service time needed to meet all the services (travel time plus waiting time).

There is a key step in the process that will result in two versions of the heuristic. Given the number of services, its very likely that there are ties among several candidate combinations. This is specially probable in the first iterations of the algorithm and therefore, the mechanism chosen to break them might have an impact on the final result.

The structure of M in an intermediate situation like that is illustrated in figure 7. There, the merger of clusters 5 and 6 entailed updating the cost of all the combinations of the remaining clusters with the new combined entity and labeling all the combinations with C_6 as non-feasible. As a result, the search for the best merger candidates returns two possibilities $C_2 + C_1$ and $C_8 + C_4$, both with a cost of 10.

Under these circumstances, the first solution that we test is making a random selection among the elements that share the same minimum cost (please note that the lower the figure, the more attractive is the resulting cluster).

The second alternative entails favoring the aggregation of clusters that seem more difficult to combine. This means that the algorithm will try to use first those clusters that, despite sharing the same minimum cost among their feasible mergers, have higher relative values when we combine them with others. This could be interpreted as a way to prevent future poor combinations. More specifically, for all the clusters involved in the merger alternatives that are part of the tie, the algorithm reviews all the potential combinations and computes their mean cost (average of the elements in the row/column in the cost table corresponding to feasible mergers). At this point, the algorithm picks the alternative that involves using the cluster with the highest value.

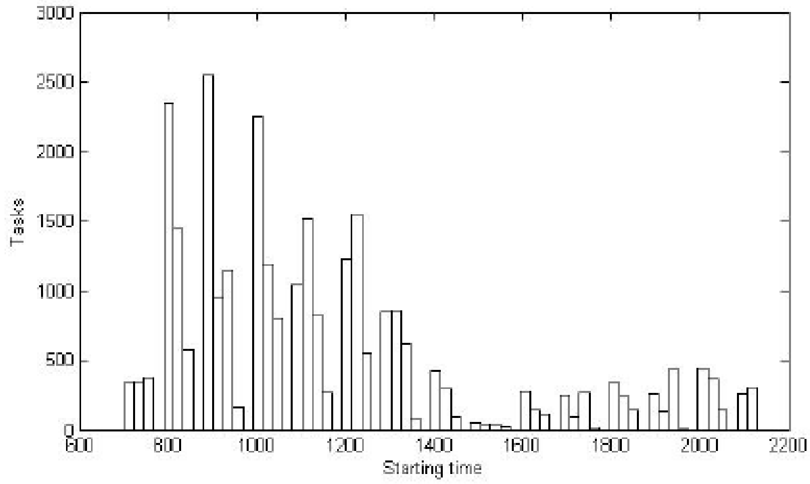

Fig. 8 Task Start Time Histogram

In the example from figure 7, the clusters involved in the tie are C_1, C_2, C_4 and C_8. The average attractiveness of the feasible mergers for them are 24, 43.3, 35 and 30 respectively. This means that it would select the merger that has C_2 among their components, that is, $C_2 + C_1$.

The performance of these alternatives on the mentioned instance is tested in the section that follows.

4 Experimentation

In this section we test the approach described before and benchmark the solution against the original resource allocation provided by the company. As we discussed in the introduction, the company is currently suffering the burden of an allocation plan that results in very poor efficiency. The reason for this is that they are currently relying heavily on full-time caregivers to cover services that have a very uneven distribution during the day. The company is adding a lot of resources to meet the services required in the 8:00 to 10:00 time bracket that end up being idle for a relevant proportion of their working day. This problem is illustrated in figure 8, that shows the number of services starting at different times during the day.

An extreme measure like hiring a caregiver per service would be optimal from efficiency's point of view, but adding extra contracts would make the company incur in additional costs. Given this situation, it is likely that a smaller set of full time employees together with a large set of caregivers with part time jobs would reduce total idle time and enhance productivity while, at the same time, controlling the administrative costs. That will be the guiding principle that underlies the chosen approaches.

We compare four different solutions. The first one will be the resource allocation provided by the company. Then, we report the results obtained by Ward's

Table 4 Metrics

	Clusters	Travel T.	Wait T.	Work T.	Efficiency
Company	1527	197424	414627	2936275	5.06
WARD	3820	148145	377844	3040604	5.78
Heuristic 1 (Av.)	3882	187552	197678	2897289	7.53
Heuristic 2	4051	113069	109860	2733128	12.26

clustering algorithm and the clustering heuristic with the two tie breaking strategies described in the previous section.

Ward's method used as benchmark is based on the well known clustering algorithm [23]. In this case, the method will merge clusters with minimum weekly accumulated traveled and waited times.

Out of the last three, only the first version of our method, labeled as Heuristic 1, is stochastic. The reason is that it breaks ties among potential cluster mergers that result on the same accumulated idle time by making a random choice. In order to control the effect of randomness, we ran the algorithm 20 times and report the main descriptive statistics of the results.

For all the above mentioned alternatives, we report five indicators. The first one, 'Clusters' represents the number of service sets or caregivers required to cover the service demand. The next two, 'Travel T.' and 'Wait T.' represent the total number of minutes per week spent by the caregivers moving from one task to the following one, and the accumulated idle time respectively. Total working time, 'Work T.', includes actual service time plus travel time and idle time. Finally, we report an efficiency measure, Eff, that is formally defined as:

$$Eff = \frac{(W_t + O_v)}{(T_t + I_t)} \tag{6}$$

Where W_t is the total working time; O_v is the overlap time; T_t represents accumulated travel time and I_t the idle time. A value of 5, for instance, in this measure means that the working time is 5 times bigger than the waiting and traveling time together, so higher values represent more efficient policies of assigning caregivers.

As we see in table 4, all the clustering algorithms result in solutions that are similar among them, but rather different from the currently implemented by the company. As we discussed in the previous section, the heuristics described in this paper have a 2.5 minutes overlap tolerance per task, but the alternative suggested by the company staff suffers from major feasibility issues as it requires major overlaps.

The solution suggested by the company requires the smallest number of caregivers. At first sight, this might seem a better solution. However, it is much less efficient in terms of total working time. Even though the number of care givers is very similar among the clustering strategies, they result in very different efficiencies. The Ward approach offers an efficiency of 5.78, substantially worse than the 7.52 average result for that metric achieved by the first version of the clustering heuristic. The main descriptive statistics related to the experimentation of Heuristic 1 are reported in table 5. There, we see that the highest efficiency obtained over the 20 executions was 7.72, close to the lowest of 7.08 which is still better than

Table 5 Heuristic 1 Stats

	Mean	Median	Std	Min	Max
Clusters	3882	3884	9.05	3866	3900
Total Travel T.	187552	187528	4020	179562	193574
Total Wait T.	197678	197865	6940	189014	221199
Total Work T.	2897289	2896704	9160	2885766	2925435
Efficiency	7.53	7.54	0.15	7.08	7.72

Ward and the initial solution. All of these are very far from the 12.26 efficiency of the heuristic with the second tie breaking mechanism. Favoring mergers of service sets that are difficult to combine during the clustering process over a purely random choice increases the efficiency of the heuristic solution by 63%. This efficiency comes at a cost of a mere 4% increase in the number of caregivers.

The reason for the increased efficiency has to do with the dramatic decrease in both travel and waiting times. This comes from the way the services are clustered. Figure 9 classifies the clusters that are part of the solutions according their accumulated working time. The solution provided by the company relied mostly on caregivers who worked more than 20 hours per week. The structure differs from the one derived from the clustering algorithms as both the Ward and the two versions of the greedy heuristic tend to favor smaller service sets.

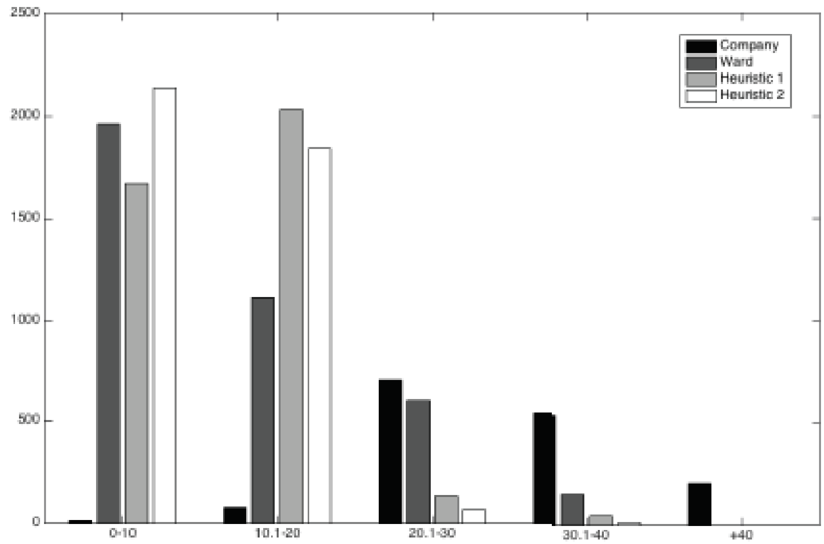

Fig. 9 Cluster Weekly Working Time Histogram

The improvements in waiting times are specially significant. The small number of part time caregivers added by heuristic with the second tie breaking strategy with respect to the Ward and the first one reduces the magnitude of this indicator by 71% and 44% respectively.

5 Conclusions and future work

In this paper we introduced two variations of a clustering heuristic that minimizes non-service time in Home Health Care Scheduling Problems. The difference between them is the way they manage ties, that is, situations where two or more clustering alternatives lead to the same efficiency.

To test the approach, we tackled a real-world instance. This problem required the allocation of professional caregivers to services that are regular in nature and must be provided at specific times during the week. The aim of the company that manages the service was obtaining alternative solutions that increased the efficiency of their former schedule. The dimension of the instance, 29,034 tasks distributed in 13,344 services, goes well beyond the standard sizes found in the literature and required an automated mechanism that we translated into the heuristic.

The strategy that we presented provides solutions that beat the efficiency of the current schedule, but the magnitude of the gain differs widely. The approach also improves a solution obtained by Ward's clustering, used as benchmark, by a wide margin. This is particularly apparent for the second version of the heuristic, the one that favors early combination of services that might be particularly difficult to merge with other clusters at a later stage. This variation doubles the efficiency of both the starting allocation and the solution provided by the Ward. allocations that require a larger number of part-time contracts.

While the company was satisfied with the balance between the efficiency and the number of caregivers required by the solutions, there is margin to improve current algorithms. Should the company provide detailed information regarding the way they balance efficiency and the number of contracts, the component of the heuristic that decides among similar merger alternatives could be fine-tuned to consider, for instance, service-set sizes among clustering alternatives. For the same reason, details on the fixed costs of adding additional contracts could also provide potential for enhancement. As an alternative, future lines of work might include a purely multi-objective approach that could provide the decision maker with Pareto fronts.

Additional paths to be explored would be solutions based on evolutionary computation to either develop new solutions or improve the output of the clustering heuristic. Others would include smart geographic partitioning of the services that could result in a significant reduction in the search space, or the implementation of parallel versions of the algorithm.

References

1. EuroStat, Population projections 2010-2060, 2011, online; accessed Apr 20, 2016. [Online]. Available: http://epp.eurostat.ec.europa.eu/ cache/ITY PUBLIC/3-08062011-BP/EN/3-08062011-BP-EN.PDF
2. A. Trautsamwieser and P. Hirsch, Optimization of daily scheduling for home health care services, *Journal of Applied Operational Research*. **3** (2011) pp. 124-136.

3. J. F. Bard, Y. Shao and H. Wang, Weekly scheduling models for traveling therapists, *Socio-Economic Planning Sciences.* **47**.3 (2013) pp. 191–204.
4. E. Benzarti, E. Sahin and Y. Dallery, Operations Management Applied to Home Care Services: Analysis of the Districting Problem, *Decision Support Systems.* **55**.2 (2013) pp. 587–598.
5. S. Nickel, M. Schröder and J. Steeg, Mid-term and short-term planning support for home health care services, *European Journal of Operational Research.* **219**.3 (2012) pp. 574-587.
6. M. S. Rasmussen, T. Justesen, A. Dohn and J. Larsen, The Home Care Crew Scheduling Problem: Preference-based visit clustering and temporal dependencies, *European Journal of Operational Research.* **219**.3 (2012) pp. 598–610.
7. J. F. Bard, Y. Shao, X. Qi and A. I. Jarrah, The traveling therapist scheduling problem, *IIE Transactions (Institute of Industrial Engineers).* **46**.7 (2014) pp. 683–706.
8. M. Mutingi and C. Mbohwa, Home healthcare staff scheduling: A taxonomic state-of-the-art review, in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)* (2013) pp.1107–1111.
9. R. Redjem and E. Marcon,Operations management in the home care services: a heuristic for the caregivers routing problem *Flexible Services and Manufacturing Journal.* in Press.
10. M. Mutingi and C. Mbohwa, A fuzzy particle swarm optimization approach for task assignment in home health care, in *Proc. IEEE International Conference on Industrial Engineering and Engineering Management* (2014) pp. 1077–1081.
11. M. Masmoudi and R. Mellouli, MILP for synchronized-mTSPTW: Application to home healthcare scheduling, in *Proc. International Conference on Control, Decision and Information Technologies, CoDIT 2014* (2014) pp. 297-302.
12. E. Cheng and J. Lynn, A Home Health Care Routing and Scheduling Problem *Technical report CAAM TR9804 Rice University* (1998).
13. J. Steeg, *Mathematical Models and Algorithms for Home Health Care Services.* Der Andere Verlag, 2009.
14. S. Bertels and T. Fahle, A hybrid setup for a hybrid scenario: Combining heuristics for the home health care problem, *Comput. Oper. Res.*. **33**.10 (2006) pp. 2866-2890.
15. P. Eveborn,P. Flisberg and M. Rönnqvist, Laps Care an operational system for staff planning of home care *European Journal of Operational Research.* **171**.3 (2006) pp. 962-976.
16. M. Rönnqvist, S. Tragantalerngsak and J. Holt, A repeated matching heuristic for the single-source capacitated facility location problem, *European Journal of Operational Research.* **116**.1 (1999) pp. 51-68.
17. C. Akjiratikarl, P. Yenradee and P. R. Drake, PSO-based algorithm for home care worker scheduling in the UK, *Computers & Industrial Engineering.* **53**.4 (2007) pp. 559-583.
18. A. Rendl, M. Prandtstetter, G, Hiermann, J. Puchinger, and G.R. Raidl, Hybrid Heuristics for Multimodal Homecare Scheduling, in *Proc. International Conference on Integration of Artificial Intelligence and Operations Research Techniques in Constraint Programming.* (2012) pp. 339–355.
19. G. Carello and E. Lanzarone, A cardinality-constrained robust model for the assignment problem in home care services, *European Journal of Operational Research.* **236** 2 (2014), pp. 748–762.
20. P. A. Maya Duque, M. Castro, K. Sörensen and P. Goos, Home care service planning. The case of Landelijke Thuiszorg *European Journal of Operational Research.* **243**.1 (2015) pp. 292–301.
21. R. Levary, An Adaptive Nurse Home Care Scheduling System, *Applications of Management Science.* **17** (2015) pp. 149–161.
22. M. Cattafi, R. Herrero, M. Gavanelli, M. Nonato and F. Malucelli, An application of constraint solving for home health care, *AI Communications.* **28**.2 (2015) pp. 215–237.
23. J. H. Ward, Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association.* **58**.301 (1963) pp. 236–244.